Collins IDEAL

Feint 418

Programming Journal
1995 Rouge Ideal

By Charlote Greenwood

Date written from 1995 to 1997

TCP/IP Network Files

\PORTS - Connections to clients

\USERS - Finger Information + Directories

\HEADS - Client Information

\IP - Address Resolution

DNS.DAT - Domain Name File

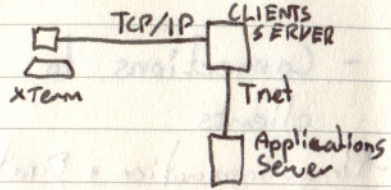
ARP.DAT - IP → Ether address

\DIN - INPUT Data Files for Ports.

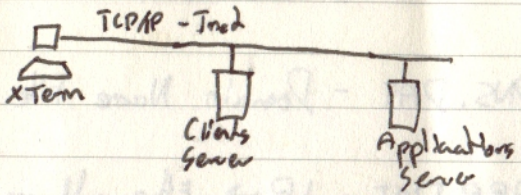
~~\DOUT - Interrupts~~

\DOUT - Received Data (Port files)

Basic Configuration



Shared Line

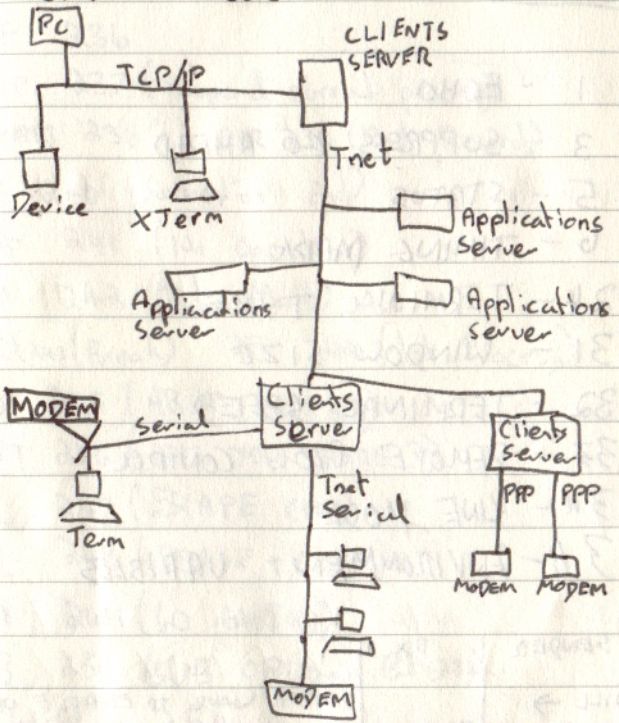


Whereby: xterm → talks to → CS

CS → talks to → xterm & AS

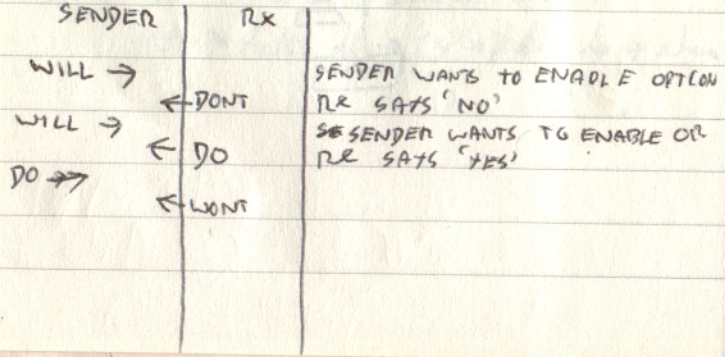
AS → talks to → CS → talks to → xterm

TCP/IP Model



OPTIONS

1 - ECHO
3 - SUPPRESS GO AHEAD
5 - STATUS
6 - TIMING MARK
24 - TERMINAL TYPE
31 - WINDOW SIZE
32 - TERMINAL SPEED
33 - REMOTE FLOW CONTROL
34 - LINE MODE
36 - ENVIRONMENT VARIBLES



Telnet Codes

EOF 236
SUSP 237 (Suspend current process)
ABORT 238 EOR 239 (End Record)
SE 240 (Subcodtion End)
NOP 241 (No Operation)
DM 242 (Data Mark)
BRK 243 (Break) IP 244 (Int. Process)
AO 245 (ABORT OUTPUT)
AYT 246 (ARE YOU THERE?)
EC 247 (ESCAPE CHR)
EL 248 (ERASE LINE)
GA 249 (GO AHEAD)
SB 230 (SUB OPTION BEGIN)
* WILL 251
* WONT 252
* DO 253
* DONT 254

IAC - 255

PORT FILE STRUCTURE

10 - {4 BYTE} NEXT EXPECTED SEQ # FROM CLIENT

5 - {4 BYTE} LAST ACKNOWLEDGED SERVER SEQUENCE NO.

9 - {4 BYTE} LAST RECEIVED CLIENT SEQ #

13 - {2 BYTE} MSS (MAX. SEGMENT SIZE)

15 - {READ FLAG 1BYTE} 0 = Not Reading 1 = Reading

16 - {1 BYTE} 0 = Nothing 1 = Data ready since last read 2 = RESET END OF READ

17 - {1 BYTE} 0 = [illegible] 1 = TX Buffer is empty.

15 - Same ACK Count {TX Buffer}

18 -> Telnet Term ID {[1] + Data}

16 & 17 (FLAGS) - Telnet Negotiation FLAGS.

A0 = Tnet FLAGS (1 = Connected 0 = Not Connected)

PORT FILE STRUCTURE

TCP INTERNAL STUFF

35 - {5 BYTE} SEQUENCE NUMBER FOR NEXT PACKET - ACK NUMBER TO WAIT FOR.

40 - {9 BYTE} TIMER OF LAST SEND (Used to check against TIMER)

49 - {5 BYTE} LENGTH OF PACKET (DECIMAL)

55 > {???} PACKET DATA (LAST SENT)

54 - {1 BYTE} 0 = NOT YET ACK. 1 = ACK

For [illegible] (35) and (9) are compared

TIMERS.DAT

(Time outs for TCP Packets)

Each record is scaned in turn when Tout occurs the packet (Row) is read from its 'PORT' file and set, the timer is then updated.

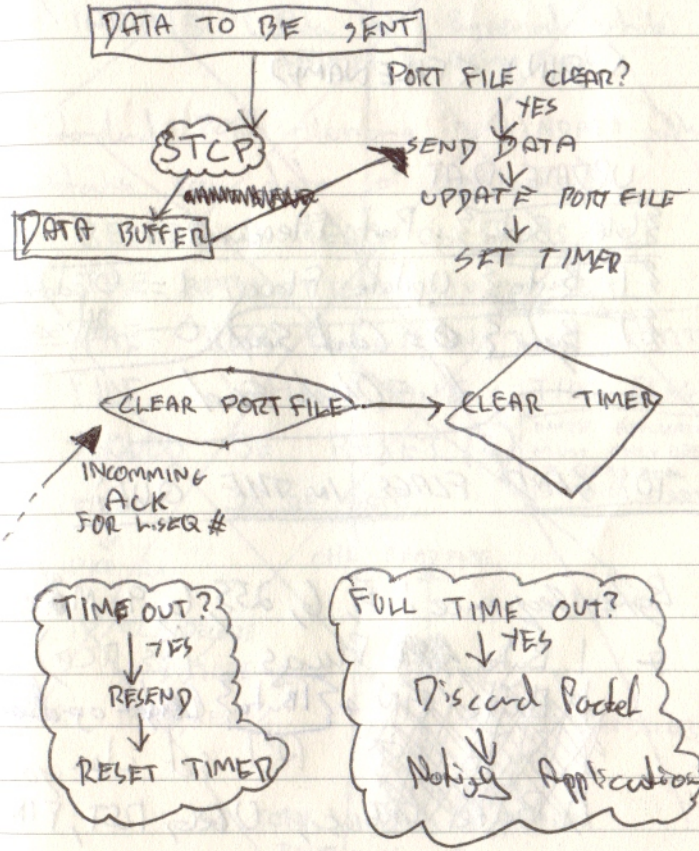
Note timer value of 000 is a 'free record'.

9 BYTE	64 BYTE	1 BYTE
Timer Value	Filename of PORT file	COUNT

'PORT' file data (the packet is read from position 55 (Length of 49EE)).

IF COUNT = somevalue then packet is discarded and application gets an error.

TCP Data Transmission



TCP APPLICATION

DATA INPUT BUFFER

<FILENAME AS PORT>

\DIN\<FILENAME>

UPDATE.DAT

{64 Byte} Port Filename

{1 Byte} Update Flag 1 = Updated

{1 Byte} 0 = Can't Send 0 = All sent

1 = Ok to Send

TO SEND FLAGS IN THE QUE:

Byte sequence: 5, 6, 255, 6, 9, Max

& 1 Byte Flags (AS TCP)

1 Byte FIN + {1 Byte} Length of data

Bits set when go.

1 Byte URG, RST, FIN

TCP READ OPERATION

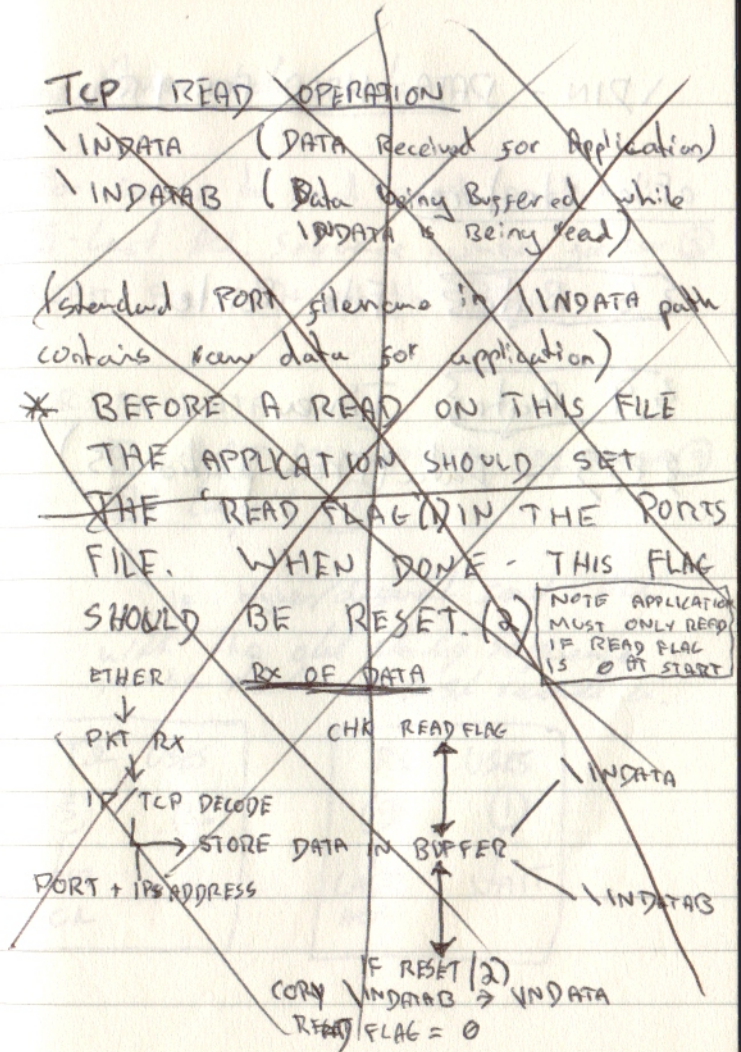
\INDATA (DATA Received for Application)

\INDATAB (Data being Buffered while \INDATA is being read)

(standard PORT filename in \INDATA path contains new data for application)

* BEFORE A READ ON THIS FILE THE APPLICATION SHOULD SET THE 'READ FLAG' IN THE PORTS FILE. WHEN DONE - THIS FLAG SHOULD BE RESET. (2)

NOTE APPLICATION MUST ONLY READ IF READ FLAG IS 0 AT START



\DIN - DATA HELD FOR TRANSMISSION

oFile structure

{6 Byte} File Pointer

{9 Byte} Timer

{???} ~ Data (From position 15)

Acknowledgements

oFor server to send next packet

[A] - Last Ack. Sequence number for sev (5)

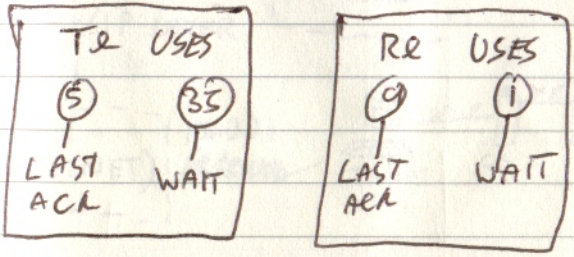
WAIT FOR [A] + Length (35)

oFOR RX DATA

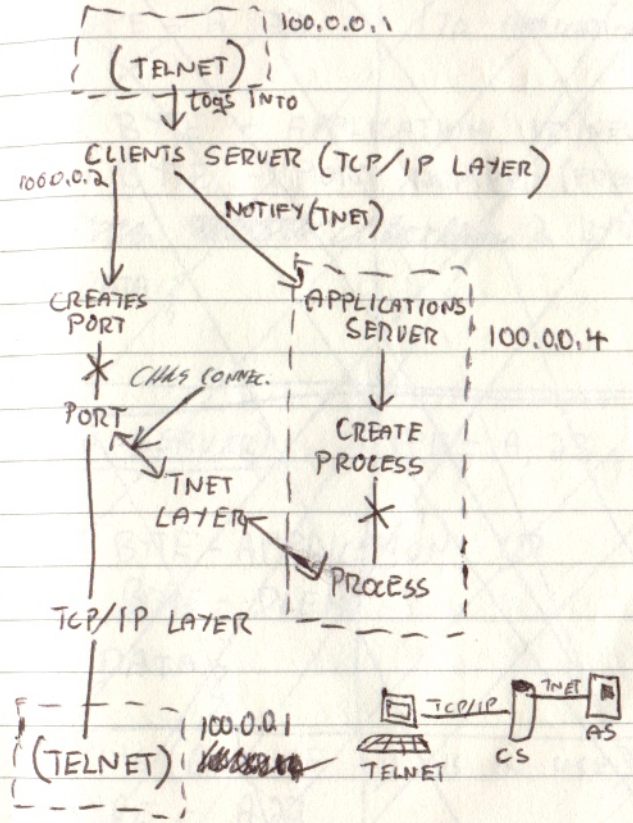
[A] = LAST ACKNOWLEDGED Seq. Number (9)

WAIT... [A] + Length (1)

↳ Is higher/different send Ack with the old starting sequence number that was last received.



Telnet example session



TNET HEADER

TYPE = A, 22 (TO APPLICATION)

2 BYTE - APPLICATION ID (NEW PER SESS)

2 BYTE - PORT NUMBER (FROM TCP/IP)

~~DATA~~ ~~HEADER~~ ~~CHECKSUM~~ 2 BYTE (DLEN)

{DATA}

(TO SERVER) TYPE = A, 23

2 BYTE - APPLICATION ID

2 BYTE - DLEN

{DATA}

PING (CHECK IF ID IS IN USE)

TYPE = A, 22

2 BYTE - APPLICATION ID

00 SERVER GETS APPLICATION NUMBER AND PORT NUMBER AS A TNET REPLY

Tnet Information

- Process ID

This is structured as follows:

{1 BYTE} — Clients Server ID (1 → 255)

{2 BYTE} — Application/Port ID (1 → 65536)

So, Max # of Clients Servers : 255
Max # of Processes per server 65536 (at any one time)

TNET

Headers

TYPE: A,22 (FROM SERVER → CLIENT)

{3 BYTE} – PROCESS ID
{2 BYTE} – PORT FROM TCP/IP
{2 BYTE} – LENGTH OF DATA
$$$$~ – DATA (Rxed)

TYPE: A,23 (FROM CLIENT → SERVER)

{3 BYTE} – PROCESS ID
{2 BYTES} – DATA LENGTH
$$$$~ – DATA (FOR TX)

- A [redacted] Tnet datagram is sent to start the connection with DLEN = 255,255
- To end connection DLEN = 255,0 ~~and client must wait for response from server.~~
- To clear data input buffer DLEN = 255,1
- For client to recieve more data – just send any datagram with proper DLEN.

INET MAP FILE (INET)

(USED TO MAP PROCESS IDS TO PORT FILES)

{3 BYTES} - PROCESS ID

{64 BYTES} - DIN FILENAME & PATH

{64 BYTES} - DOUT FILENAME & PATH

000...0 = FREE RECORD

D → D (Device to Device)

255 (Always at start)

OPCODE BYTE

~~DEVICE BYTE~~ 2 BYTE ADDRESS (FROM)

2 BYTE ADDRESS (TO)

Data Len HB, Data Len LB

{Data}

① write data to screen (text)

{X Pos} {Y Pos}

{Colour} {CHR}

HB: Backg. LB: Pen

Automaticly line wraps, at y=25 need: 1.

② Sound speaker

{Duration (150)} {Tone}

③ Select Mode

{Mode}

Selects screen mode. 0=text, 2, 4 - 320x200, 640x200

④ Image Download (1) - Course, line art

{X pos} {Y pos}

{Length & Pixel}

HB LB

Pixel = Colour (2 Bit)

Length = Total pixels to colour (6 Bit) 0=1 pixel

⑤ Image Download (2) - Fine

{X pos} {Y pos}

HB		LB	
Length	Pixel	Length	Pixel
2 BIT	2BIT	2BIT	2BIT

As above but 2 Bit version for the run length.

⑥ Clear screen

⑦ Monitor On

⑧ Monitor Off

⑨ Modem Line Control

{Func}

1) = Pick up line
2) = Drop line
3) = Login User
4) = Send message data
{message}
5) = Logout User
6) = Goto automatic Offline user interaction.
7) = Goto online user interaction.
8) = Patch Number
{Patch}

(10) Patch Data (From Device)
{Signal} {Data}

(11) Store Binary Image to Disk
{Block #} 0 creates file, else it appends
{Data} (Need {64} = Path + Filename)

(12) Patch Interactive Automatic Mode to File
{Filename} (Local)
· Runs the EEE code when user is online.

(13) DTMF Code Signal (From Device)
{Button Press}

(14) Play MP3 File (IN /Sounds)
{File Name}

(15) Erase File
{64 Path + Filename}

(16) Report Files in Path (INC SPACE)
{Path}
({Signal}
Signal ~~attributes~~ is the data's handle (Func 10 returns)

(17) Device Control
{Device #} {Signal}

(18) Reset Device Control

(19) Read Device (IR sense) Lines
{Signal}
· Returned in func 10

(20) Set telebord Signal
{Signal}

(21) Set telebord condictions
{Condiction do retorno}

(22) Append ARP List
{IP} - {Ether}

(23) Delete ARP
{IP} - {Ether}

(24) Append Nslookp
{IP} - {Name address}

(25) Delete Nslookup
{IP} - {Name address}

(26) Send Total ARP store
{Signal}

(27) Send Total Nslookup store
{Signal}

(28) Full Reset

IP

0		16	31
4 BIT Ver	4 BIT HEAD L	8 BIT TOS	16 BIT LENGTH
16 BIT ID		3 BIT FLGS	13 BIT FRAG OFFSET
8-BIT TTL	8-BIT PRO	16 BIT HEAD CHKS	
32 BIT SOURCE ADD			
32 BIT DEST IP ADD			
OPTIONS			
} DATA {			

TCP

0	16	31
16 BIT SOURCE PORT	16 BIT DEST PORT	
32 SEQ. #		
32 ACK #		
4-BIT H-LEN / RES 6-BIT / URG / ACK / PSH / RST / SYN / FIN	16 BIT WINDOW SIZE	
16 BIT CHKSUM	16 BIT URG P.	
OPTIONS		
} DATA {		

offsets **GLOBAL TCP TABLE**

	Source P.	Dest. Port (+2)	
+0	{2 Byte}	{2 Byte}	
+4	Seq # {4 Byte}	Ack #	(Receive) PORT
+8	Seq # {4 Byte}		(Transmit)
+12	{8 Byte}		(IP PATH)
+20	{2 Byte}	MSS	
+22	{1 Byte}	LINK Condition (128 NEW, 64 FIN, 32 RST, 16, 8, 4, 2, 1 PARENT)	
+23	{2 Byte}	Time Out for Retransmission	
+25	{1 Byte}	Retransmit Count (Set to 0 when Ack)	
+26	{2 Byte}	Window size	90 Bytes Total
		Application Data	
+28	{? Byte}		

(Filename = /Ports/(Hex IP address))

(Need Record = +90)

+81 {8 Byte} Get data Filename. (.out .IN .mal etc)

(29) = 0 = Options Not checked
1 = options checked

Telnet Options Storage

+30 Echo (1)
+31 Suppress go ahead (3)
+32 Status (5)
+33 Timing Mark (6)
+34 Terminal Type (24)
+35 Window Size (31)
+36 Terminal Speed (32)
+37 Remote Flow Control (33)
+38 Linemode (34)
+39 Enviroment Variables (36) Done

{1 Byte} = H = | 8 WILL | 4 WONT (2)
DOs + Donts (High) Low

128	64	32	16	
Must Have	Must Have Not	Try Have	Status	0 Not Neg
				1 = Neg Started
(Both 0 = Dont Care)			(1 = On 0 = Off)	~~2 [illegible]~~ Given

Buffer (Out/Te)

3 = 1510 (10)
5 = 1520

[1] 0+1: [2 Byte] Length of packet
[3] 2+3: [2 Byte] Data pointer
$$$ [5] 4+5: [2 BYTE] Next free Record
Data
$$$

17~~0~~0 = PD$ Template 34 (4x)

As standard, when started a ~~10~~ 1800 byte gap is left at the start of this file for Te packets.

Start = +1804 200 ~~[illegible]~~ ~~200~~ Byte APP temp data store

DATOUTS [1550]

1549 = Le(che)

Data =
{2 Byte} length 1510 – 1529 (…)
{1 Byte} Flags 1530 – 1549 (…)
{1 Byte} Options length
$$$ Options
$$$ Data

Telnets.dat
Fingers.dat
Mails.dat

Application Storeage (Telnet Access)

+1 {4 Byte} Pointer ID
+4
+40 {1 Byte} Internal Process FLGS
+45 {2 Byte} Data IN pointer (pointer = +1)
+47 {2 Byte} Data Length
+49 {1 Byte} Login Count
1

Application. Conf. File

Telnet Conf. / Finger Conf.

{8 Byte} Path to Buffer file
{8 Byte} Filename (Buffer file)
{8 Byte} Filename Conf. File
{2 Byte} Pointer to Rec (Rec. #)

20 Byte Records

(0000 = New/Free Record

= IP Filename + Port # ((Dest.)

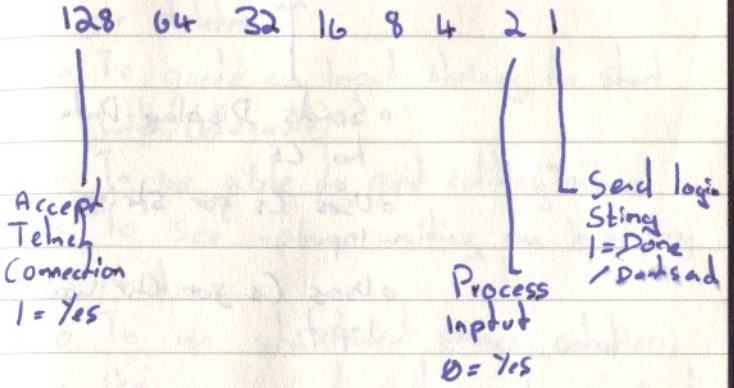
DATIN$ (FILE FORMAT)

{2 Bytes} @ $$$$
Write Position

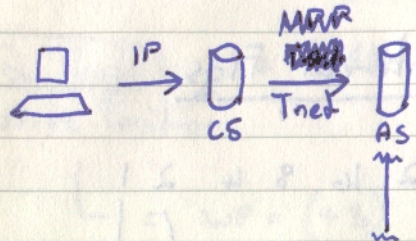
└ = WP = (+3)

Telnet Details

(rc +40) = Internal Flgs



Telnet Session

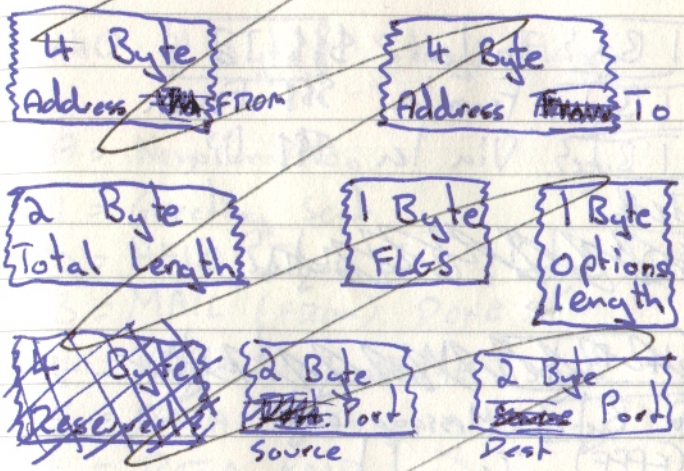


- Sends Display Data to CS.
- Uses Cs for string input.
- Uses Cs for Chr waiting. Input

CS String (INTEXT)

- Must be able to turn echo off
- To quote a chr other than Space for deletes
- To quote an input string to start with. (default)
- To be able to quit editing mode
- To set up a waiting chr in editing mode
- To use standard emacs notation.
- Use of a timeout, with asc. input.
- Define length max.
- To return when max is reached with wrap word, or to just stop.

Tned Frame



$$$ Options (0 > 255)

$$$ Data

Mail spool file

{1 Byte} To len $$$ To
{1 Byte} From len $$$ From
{1 Byte} Via len $$$ Via
{1 Byte}
[redacted]

{2 Byte} Message length
(FFFF) = Truncated
where 2 Byte = Continued Length.

eg. (one message)
[length] $$$$ [Length] $$$$
FFFF upto 7FFF (32767) FFFF
[length] $....$10 END
0010
FFFF = 7FFF (32767)
Anything above is truncated

SMTP Process Variables

+40 {1 Byte} Stage Flag

0 = New Connection
1 = Greeting Sent
2 = HELO Done
3 = MAIL (FROM) DONE
4 = RCPT (TO) DONE
5 = DATA BEGUN
6 = DATA END

At 6 Data is stored and flag is reset to (2).
Temporary storage of Data is at DATIN$ position 1600 on: using format on opposite page.
1600 = VIA Data 2000 = From Data
1800 = To Data 2200 = Message ID

~~Accounts File Formats~~

[redacted]

{FLG1B} 128 64 (+34)

128 = No Password Required

64 = Account disabled

({FLG 2B} Reserved) (+35)

{24 Byte} Password (+36)

[redacted]

(+) (After)

+60 {10 Byte} Date login | Mail Re 96
+70 {10 Byte} Date logout | Mail Read 106
+80 {8 Byte} Time login | Mail Re 116
+88 {8 Byte} Time logout | Mail Read 124
{4 Byte} last IP

TOTAL LENGTH = ~~96~~ Bytes 232

Mail File Formats

- All mail data stored in \server\Mail
- Accounts in \server\user\<Aclog>

users
<[redacted].dat>

{24 Byte} User ID (+0)

{8 Byte} Mailbox filename (+24)

{FLG1A} 128 64 32 (+32)

128 = Mail [redacted] waiting

64 = Apply Rules

32 = Unread mail

({FLG 2A Reserved}) +(33)

Mailbox = <F> + .MAL

UnRead Mail = <F> + .URM

Rules = <F> + .RUL

+136 {64 Byte} Email WHERE <F> = (MAIL)

+200 {32 Byte} Read Name

.MAL + .URM Formats

{1 Byte} length From
{1 Byte} length Via
{B Byte} length To

{2 Byte} Message length
($FFFF) = Truncated
{16 Byte} Message ID (POP)
{10 Byte} Received Date
{8 Byte} Received Time

$$ From
$$ Via
$$ To
$$ <Message> upto ($FFFF) (65,267)

{1 Byte} Message length 254

If Message is truncated ($FFF len) then after message is another 2 Byte len.

Mail Rules

{FLG} 1 Byte (<F>+.RUL)
1 = Extended. See onwards
Lin server/mail/<F>/?

128
' Redirect all to

64
' Redirect from <s7 to <s7

32
' Delete all from <s7

16
' Delete all

8
' Mail document <s7 (/serve/mail/does) back on <s7

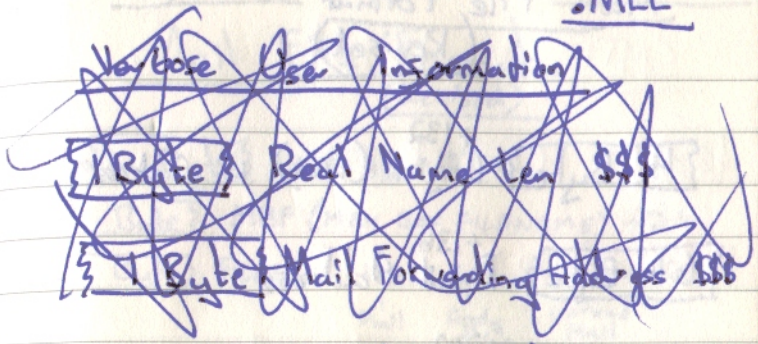
{1 Byte} Argument length
{Argument} $FF
' Data

Mail Forwarding

{1 Byte} $$ To Address
{1 Byte} $$ From Address
{4 Byte} IP Address (TO)
{1 Byte} Retry Count
{1 Byte} + $$ ... Message

(MFORWD.DAT
(SERVER/MAIL/<F>)

.MLL



.MLL Mail Listings File

{10 Byte} Message Length
{18 Byte} Message ID
{10 Byte} .MAL Record ID
{1 Byte} Flag (POP)

Used in POP3 Access

Flag 1 = Deleted (waiting Deletion)
2 = Read

Mail File Format (Revised) USERS.DAT

{4 Byte} +132 Login (LAST) IP Address

{64 Byte} +136 Email Address

{32 Byte} +200 Real Name

<-> 232

{24 Byte} +232 POP3 Password

<-> 256 {24 Byte} Group <-> 280

(for FTP set POP3 to 'ftp'; Real Name
= Rights (R-Read, W-Write, L-List) Email
= Path to access (root) -Use /

File = /SERVER/IP/[redacted] USERS.DAT

*USERS.DAT

Mail File Format (Revised)

{24 Byte} USER ID +0

{8 Byte} MBF (MAIL BOX FILENAME) +24

{1 Byte} FLG1A 128 Mail, 64 Apply Rules, 32 Unread Mail +32

{1 Byte} FLG2A +33

{1 Byte} FLG1B 128 No Password Required, 64 Account Disabled +34

{1 Byte} FLG2B +35

{24 Byte} Password +36

{10 Byte} Date Login +60

{10 Byte} Date Logout +70

{8 Byte} Time Login +80

{8 Byte} Time Logout +88

{10 Byte} Date Mail Received +96

{10 Byte} Date Mail Last Read +106

{8 Byte} Time Mail Received +116

{8 Byte} Time Mail Last Read +124

SMTP Multiple Recpts

(DATING OFFSETS)

[3000] + <...>

{1 Byte} {...} terminated by ∅
RCPT
Length

eg.
[4] root [3] Mag [0]
sends to root and Mag @ 'domain'
Cap to 255 domains
(RC+42) = # of (RCPTS)

(RC+43) {2 Bytes} Pos of next (RCPT)
(RC+45) {2 Bytes} Internal pointer

WEB (HTML) ACCESS

Startup Flags
+40 128 64
| Get <filename> stored
| Transmission Ready (1)
(Reset to ∅ when done (0)=DONE)

Tnet Frame

{8 Byte} Destination ID

{8 Byte} Sender ID

{4 Byte} Data length

$$$ Data

{Ethernet Type = 2,2
Dest ID = FF.FF.FF... = Request Connect
Dest ID = 00.00.00... = Disconnect
Data Length = TCP Port service (2) Fixed.

RC+40
0 = New con
1 = Greeting sent
2 = User Done
3 = Pass Done

POP3 Access

(DATING)

+2000 [<256] User ID
+2300 [<256] User Password
+2460 [24] POP3 Password

Eterminal Access

+40 {1 Byte} Internal connection status

0 Nothing sent/established
1 Options/setup sent
2 Configuration Received
128 Ready for transfer.

+41 {8 Byte} Tnet AS ID
+44 {8 Byte} Local Tnet ID

@INI$ stored in MAIL@$

+57 IMAGE / BITMAP ORDER

1 IMAGE 2 BITMAP

(0=LSB FIRST / 1=MSB FIRST)

SEE REVISION

Tnet => Internal Lookup

{2 Byte} Tnet SID entry check

+4 {10 Byte} 'RC' file locator

+14 {1 Byte} Flags (1=Valid) (2=WAIT STATE)

+15 {8 Byte} Configuration filename (def. CNF do F$)

+23 {6 Byte} Either of remote Bytes Valid.

(Tnets.CNF) in TCPPATH$

S.PORT 2 BYTE 29

R.PORT 2 BYTE 31

S.IP 4 BYTE 33

Bytes Length

Tnet Options

*1 = Sync Seq# (sends each time)

*2 = Acked + {2 Byte} seq #

*3 = Send Last ACK

*4 = Close connection

*5 = Connection closed.

*6 = New Connection

*7 = Connection Accepted

*10 = Set Tnet Incomming route
+ {2 Byte} (IP Tnet Node - 72.72)

11 = Unset Tnet Incoming route
+ {2 Byte} (IP Tnet Node - 72.72)

*50 = ERROR + {2 Byte} Seq. # where error was.

50 = ERROR + SEQ + opcode

Tnet - TCP/IP options

< 100 = State TCP/IP Connection Source

{2 Byte}	Source TCP port
{2 Byte}	Dest. TCP port
{4 Byte}	Source IP
{4 Byte}	Dest. IP
{6 Byte}	Source Ether. Address
{1 Byte}	IP Protocol (20 Byte)

Note: when a new IP connection is detected for Tnet, this datagram is broadcast on eth FF. Tnet servers check the ports/IP, and except by sending (7) Connection Accepted. If connection has already been accepted by another server, OP (4) close connection is sent to the server.

Tnet - TCP/IP Startup

101. State TCP/IP Connection Desd.

{2 Byte} Dest Port
{4 Byte} Dest IP
{1 Byte} Protocol

{1 Byte} Length of TCP options NBi FOR ICMP
$$$ options OPTIONS ARE NOT USED INSTEAD: {1 Byte} {1 Byte} Type Code

NB: Protocol must equal TCP (6). For comdios. UDP (17) can be supported but only data in this frame is sed.

TCP=6, UDP=17, ICMP=1

Tnet > TCP.IP

~~102 = Connection on TCP/IP OK~~
~~103 = TCP/IP connect or refused/tem~~
104 = Disconnect from TCP/IP
105 = Set TCP flags + {1 Byte}

110 = Clear Incomming Buffer
111 = Priority - dont que.
112 = Clear Outgoing Buffer

120 = Get ALL for IP
125 = Get ALL fo Ethernet

Inet: Autosend

200 = Program Autosend

{4 Byte} - Handle

{1 Byte} - # of conditions

(
Conditions:

{2 Byte} position in Data

Compare to = {1 Byte} len $$$
)

201 = Autosend OK + {4 Byte} Handle

202 = Autosend Refused + {4 Byte} Handle

205 = DELETE Autosend + {4 Byte} Handle

206 = DELETE All Autosends

NB: Autosend requires TCP connection to be accepted first.

Inet - Autosend

210 = Program Autosend {DATA}

{4 Byte} - Handle

{2 Byte} + Data $$$

211 - Clear Data in Autosend

{4 Byte} - Handle

212 - Get Autosend Handle Stats

{4 Byte} - Handle

213 - Autosend stats

{4 Byte} - Handle

{2 Byte} - Total Hits

214 - Set Autosend Notify + {4 Byte}

215 - Clear Autosend Notify + {4 Byte}

216 - Autosend Notify + {4 Byte}

Tnet R. Login

150 = Login user
+ {24 Byte} USER ID
+ {24 Byte} PASSWORD

(

Printer

155 = Send to Printer
+ {1 Byte} PORT

Sends all data that follows to the printer.

151 = Get USER DATA
+ {24 Byte} USER ID

* Returns the USERS Record (152) otherwise (153), Mail/Extended Mail rules are also returned if any exist.

({USER REC} $$$ MAIL RUL $$$) "152 Return"

FILE FORMAT
INET.WTT

Tnet. Request IP File

{2 Byte}	SOURCE PORT	0,0,0,0,0 = FREE REC
+2 {2 Byte}	~~Dest~~ REMOTE PORT	
+4 {1 Byte}	PROTOCOL	
+5 {4 Byte}	Remote's IP Address	
+9 {2 Byte}	Tnet SID (FROM SERVERS POINT OF VIEW)	
+11 {2 Byte}	Tnet DID (REMOTE)	
+13 {6 Byte}	Ether Tnet Route	
+19 {10 Byte}	Time out timer	

29 Bytes

* Used for checking of an incoming IP connection is for a remote Tnet route. Timeout is after 25 secs. Record is erased when found/TO.

{LM TCPPATH#}/TNET.WTT

FILE FORMAT
TNET.CNF

Revised

TNET INTERNAL Lookup

{2 Byte} Tnet SID 0,000 = FREE REC
+2 {2 Byte} Tnet DID

+4 {10 Byte} 'Rc' file Locator

(INTERNAL CONFIG PORTS)
+14 {1 Byte} Flags. (1=VALID) 32 = PORT 1
+15 {8 Byte} 'CNF' filename root 64 = PORT2
+23 {6 Byte} Tnet route ether 8 = PORT3
(IP) 4 = PORT4
+29 {2 Byte} Source Port
+31 {2 Byte} REMOTE Port
+33 {4 Byte} Remote IP
+37 {1 Byte} Protocol

38 Bytes

(IN TCPPATH1) / TNET.CNF

Tnet.Pro Process Table

Tnet Process (AS)

10,00 = Free Rec

{2 Byte} Local Tnet Port
+2 {2 Byte} Remote Tnet Port
+4 {6 Byte} Remotes Ether

+10 {4 Byte} Process Handle / Type
+14 {4 Byte} Index
+18 {4 Byte} Sub. Index

+22 -> $$ {64 Byte} Internal Process Data

86 Byte

Pre-Defined Handles (0 => 255 Reserved)

1 = Gopher 2 = WWW
3 = Std. Telnet 4 = Finger
5 = FTP 6 = Archie

256 = MouseSound 257 = MouseVideo

Extended Mail Information

(F.2) WHO IS Information

{1 Byte} len {$$$} Organisation
{1 Byte} len {$$$} Organisation Type
{1 Byte} len {$$$} Work Tel.
{1 Byte} len {$$$} Fax #
{1 Byte} len {$$$} Department
{1 Byte} len {$$$} Address 1
{1 Byte} len {$$$} Address 2
{1 Byte} len {$$$} Town/City
{1 Byte} len {$$$} Country
{1 Byte} len {$$$} Postcode
{1 Byte} len {$$$} Country
{1 Byte} len {$$$} Cc'ed Email

FTP FLAGS	128	64	32	16	1
	ADMIN	UPLOAD	DNLOAD	NO CHANGE DIR	GOD LINE

Extended Mail Information

1st Byte
{Set bit 0 to 1 in .DUL file}

{Extended Function} 1 byte (F1, F2...)

{2 Byte} Length (Fe. above 3 bytes)

(F.1) Set FTP rights and paths

{1 byte} PATH Length
{$$$} PATH (BASE PATH)
{1 byte} PASSWORD Length
{$$$} Password
{1 Byte} GROUP Length
{$$$$} Group
{1 Byte} Flags + {$$ Byte} MAX FILE
+{1 Byte} Into Line {1 Byte} SIZE (TOTAL)
+{1 Byte} FTP User + {$$} SIZE (K)

Extended File Inf. Block

PERMISSIONS / TYPE

{1 BYTE}

{1 BYTE}

WORLD
[4 2 1]
R W X

TYPE
[2^7]
D

GROUP
[32 16 8]
R W X

OWNER
[128 64]
R W X

+62 {1 Byte} is used during a directory update for bagging

Extended Paths
{2 Byte} RFLG
{260 Byte} PATH
260

Extended File Information Block

FIB
~~FIB~~

ΔEFIB.EXT = Extended Paths

ΔEFIB. ~~EFI~~ (127) (Data file resides in current directory, regarding do the files in it.)
(127)

(463)
{FLAGS} 1 = Don't Display Paths
2 = Extended NT Path

{2 Byte} FILE PERMISSIONS/TYPE
+2 {2 Byte} # of Linked filenames (Files in directory)
+4 {24 Byte} owner
+28 {24 Byte} GROUP
+52 {8 Byte} SIZE (DIRECT in Bytes)
+64 {8 Byte} TIME Modified {8 Byte} Reserved
+72 {8 Byte} TIME created
+80 {10 Byte} DATE Modified
+90 {10 Byte} DATE Created
+160 {128 Byte} FILENAME
+228 {12 Byte} REAL FILENAME
240 Bytes

EFIB PATHS Information

FILE

{255 Byte} PATH
{4 Byte} Total Files in path
{5 Byte} Total Space Used
{8 Byte} Time last modified
{8 Byte} Time created
{10 Byte} Date last modified
{10 Byte} Date created
{2 Byte} Reserved
{24 Byte} Owner
{24 Byte} Group

FTP Flags

128	64	32	16	1
ADMIN	UPLOAD	DNLOAD	NO CH DIR	GOD LIKE

4600 Rename From

FTP + EFIB

FTP - Vuibles (INTERNAL)

(USER) 5000 EFIB 'LONG' PATH NAME

4000 DEFAULT (BASE) DIRECTORY

4300 CURRENT DIRECTORY

↑ STORED FROM INITIAL LOGIN.

4600 GROUP (FTP LOGIN)

4650 FTP FLAGS

4660 FTP MAX SIZE UPLOAD

4700 MAIN LOGIN ACCOUNT GROUP
4760 FTP V. USER

FTP Vuibles Returned By 'Getuser'

Search: ($$ = Login / FTP$ = Pass)

- EFIBPATH$ - Default Path
- FTPGROUP$ - Group for FTP login
- FTPMSIZE$ - FTP MAX SIZE
- FTPFLG - FTP Access Flags (0) = NO FTP
- FTPINFO$ - Information line
- FTPPASS$ - FTP Password
- FTPUSER$ - Virtual Username

INTERNAL RC VARIBLES

① GLOBAL

RC+30 = Telnet Mask (Multi)
RC+40 = Telnet Flag (MAIN)
RC+81 = Filename Root
RC+29 = Telnet Flag
RC+45 = Telnet DATIN Marker
RC+49 = Login Count (Telnet)
RC+55 = Tnet Connection Info + TIMER
RC+54 = Tnet Rx. Sequence

RC+50 = FTP (IP + PORT) OF CLIENT
RC+41 = FTP/GENERAL REGISTER

RC+70 = IP + PORT to listen for.
RC+56 = FTP FLAGS (1 = Passive Connection)

Revision 2

FTP Extended Data

{1 Byte} + $$ Path
{1 Byte} + $$ Password
{1 Byte} + $$ Group
{1 Byte} (Flags)
{1 Byte} + $$ Max Size
{1 Byte} + $$ Information Line
{1 Byte} + $$ ~~Username~~ / Login

RC+69 = PASV USE TO. 0 = No PASV
RC+68 = TYPE (I or A)

Extended HTTP
Path information

51 Bytes: Path length
999 Path
51 Bytes: Group length
999 Group (only has

51 Bytes: Flags

NB: Must have FTP access as well as HTTP.

Connection is refused on any port to which there is no access.

DNS Extended

To the DNS database file the user login string of 24 Bytes (fixed) is added to the end of this file.

=>

$$DNS => ...

524 Bytes: USER ID

For HTTP/FTP these will be looked up in the extended USER INFORMATION to obtain their values.

Note: IP can be any address however other must = the host which has access to the user files.

DNS.DAT - File Format

EFIB Notes on V.Paths

The Extended File Information Block (EFIB) method for extended paths:

EFIB DPATH

- DOS1 — DOS2
- EFIB1
- EFIB1/EFIB2
- EFIB1/EFIB3

All directorys are stored from dos root.

NB: The '..' entre contains the directories parent path if that directory is of the extended format. (IN FL$)

JUL 97

Mail Serv

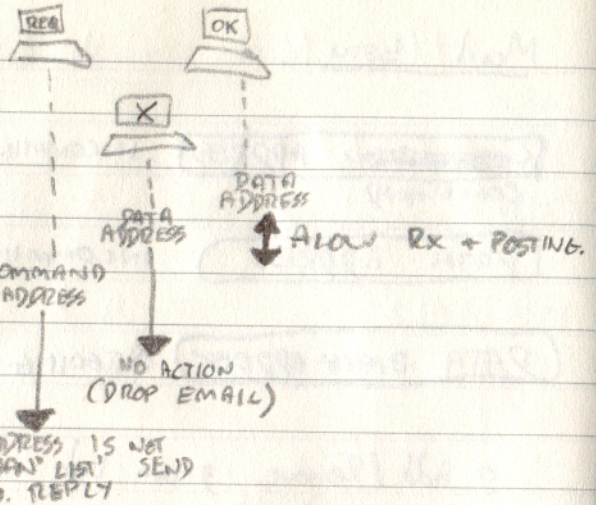
~~SUBSCRIPTION~~ ADDRESS — INCOMMING COMMAND

DATA ADDRESS — INCOMMINE

DATA BATCH ADDRESS — OUTGOING

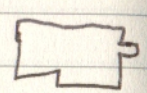
- Add / Remove from list
- ~~Dis~~ Provide a range of requested files
- Maintain a subscription list for distributions
- Alow / Disallow global mailings from users.
- provide away to email d~~i~~s. to the list serv for distribution (secure admin list) Password protected

Tmail serv


REG
OK
X
DATA ADDRESS
DATA ADDRESS
ALOW Rx + POSTING.
COMMAND ADDRESS
NO ACTION (DROP EMAIL)
IF ADDRESS IS NOT ON 'BAN' LIST' SEND APPRO. REPLY

Teletext Adapter - PINS USED

COMPONENT SIDE →



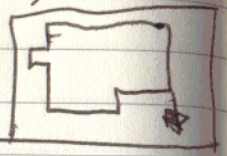
~~GND~~

I/O CHANNEL CK

D7, D6, D5, D4, D3, D2, D1, D0

I/O CHANNEL READY

A10, A1



PCB SIDE

GND, +5VDC, IRQ5, DRQ1, +5VDC, GND

Media Workstation

Network Adapter	INT 10
AWE	INT 5
WIN MOTION	

Network Adapter	300H
AWE	260H
WIN MOTION	

INTERNAL DMA = 3

AWE LOW DMA = 1

AWE HIGH DMA = 7

Actuator Data

Movemat = Blue, Green

Detect - Black (GRND)
Red 5v
Yellow Pulse

Mudw logins

LOGIN PASSWORD

[redacted]

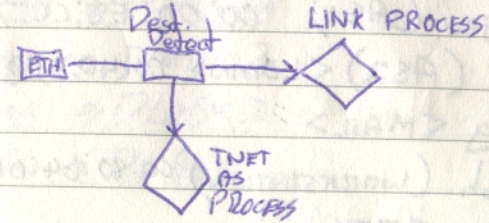
leewor

fpser 30s

£20 dp.

DUAL TNET/~~PPP~~ LINK SERVER PROPOSAL

- Time sharing Arch.



- TNET AS would be for ~~BASIC~~ APPS. (SMTP, POP3, FINGER, WHOIS)

- LINK PROCESS would be able to decode SLIP & PPP & TNET/GT (FOR MCD ETC).

Rover (UNIX) A0:FA:00:00:00:02
Bruno (CS1) 00:00:E8:CC:67:A4
Shazer (AS1) 00:00:E8:CC:71:FD
Rex (AS2) <LOGIN> 00:40:33:39:[illegible]
Fang <MAIL>
Flash (WORKSTATION) 00:80:64:04:39:[illegible]
Doze <DATA>

High Speed Serials

9600 → DLAB=12
19200 → DLAB=6
38400 → DLAB=3
57600 → DLAB=2
115200 → DLAB=1

3F8 (Low) 3F9 (High)

(3)FB BIT 7 (128) = 0 unset
1 to set/view
(Reset bit when done)

(Ports.) 3F8 com1 2F8 com2
3E8 com3 2E8 com4

A. B. C. D

IP Network stems

A(10) - Local Network
B(200) - Machine
C(0) - Type
D(1) - Number

Bruno 10.200.0.1
Bruno-Intername 10.200.32.32
Bruno-interroute 10.200.100.1

Type 32 = Name Server
100 = Router
24 = Proxy Web /web
16 = Locate Default domain
72 = Tnet
~~80 = Remote Users~~
~~80=12, 84=12, 88=12, 92=12~~

Eg. (BRUNO REMOTES)

10.200.80.1 = Remote Port 1
10.200.80.2 = Remote Port 2
10.200.80.3 = Remote Port 3
10.200.80.4 = Remote Port 4

(10.200.80.n)
- n = port Number.
- 80 = Remote Type

0 = Local Services
16 = Local Domain (tree.atlati)
24 = Web
32 = DNS
72 = Tnet
80 = FTP
100 = Router

CGA Wall Monitor Cable

Tandy (Original)	CMS pinout (Twisted)
Sheild/GND	- Brown
Red	- Orange
Green	- Green
Blue	- Blue
Black	- Brown/white
Orange	- Orange/white
Brown	- Blue/white
Yellow	- Green/white

(Microsoft STYLE)

Mem strings

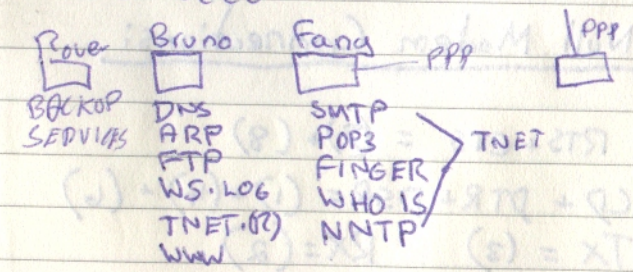
Pointer to:

{Len LB} {Len HB}

{Offset LB} {offset HB}

All in the same segment.

* Because of the huge nature of FTP the following configuration must occur:



Rcd = (1) FLAT SERIAL

(1) = P8 (CD)
(2) = P3 (RXD)
(3) = P2 (TXD)
(4) = P20 (DTR)
(5) = P7 (GND)
(6) = P6 (DSR)
(7) = P4 (RTS)
(8) = P5 (CTS)
(9) = P22 (RI)

*Null Modem Connections:

RTS + CTS = (7) + (8)
CD + DTR + DSR = (1) + (4) + (6)
TX = (3) RX = (2)
GND = (5)

C | start 0 → 81 (40.3) PRI
-- 82 → 488 (200.3 MB) AS
D | 489 → 927 (2160 MB) PRI
-- 928 → 1023 (59.5) Ext Dos (Active

EFIB 'DOS VIEW'

(EFIB ROOT)
* files
* path1 → files . path1.b
(1) path1.a → files . path2
(2) path2 → files

ROOT | 2ND LAYER |

4 PIN Extension
10 PIN OUT

(Red) (1) ~~8~~ (CD)
(2) → (3) (Re)
(3) → (2) (Te)
(4) → 20 (DTR)
(5) → 7 (GND)
(6) → 6 (DSR)
(7) → 4 (CTS)
~~(8) → 5~~
(8) → 5 (RTS)
(9) → 23

(6+8+20) = 6+1+4
(4+5) = 7+8
DOUT/IN = 3 (Te), 2 (Re)
GND = 5

VIDEO LINK

[279A]

12/13 - FB/LS (LPT 2)
11 10
LUM
10 11(INV)

[379H]

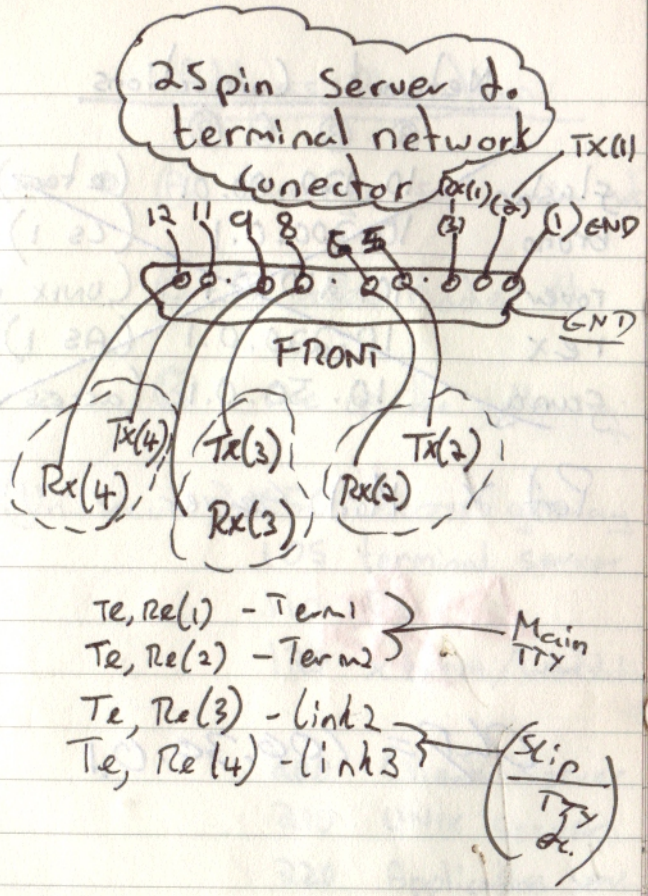
13
12
LUM
11 11(INV) (LPT 1)
10

(BITMAP)

10 → 13 = (128, 64, 32, 16).

PIN	BITMASK [x79H]
10 =	64
11* =	128* (INV)
12 =	32
13 =	16

LOW (13, 12, 10, 11) HIGH



Network Connections

	Ⓐ	Ⓑ	Ⓒ	Ⓓ	
flash	10.	120.	00.	01	(xterm)
bruno	10.	200.	0.	1	(CS 1)
rover	10.	210.	0.	1	(UNIX 1)
rex	10.	220.	0.	1	(AS 1)
fang	10.	50.	0.	1	(access 1)

Port 25 - Mail Server

OK? = 100.30.0.1

Address Allocation

- Ⓐ changes for main net
- Ⓒ + Ⓓ ~~add~~ node #
- Ⓑ one of the following:

(Un. Secure) * 50 access gateway
105 terminal server
110 PC
120 xterm/workstation

200 Clients Server
210 UNIX server
220 Applications Serv.

NULL MODEM (FULL)

25 PIN LAYOUT

1 → 1

*2 → 3
*3 → 2
} DATA Te, Re

4 + 5 CTS - RTS

*7 → 7 GND

6 + 8 + 20 DSR · CD · DTR

11 → 12
12 → 11
} SECONDARI DATA

BROADCAST PIN
072 962

NULL MODEM (FULL)

9 PIN LAYOUT

1 + 4 + 6 CD · DTR · DSR

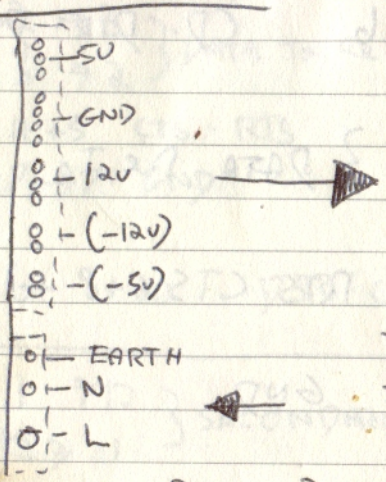
*2 → 3
*3 → 2
} DATA Re, Te

7 + 8 RTS · CTS

*5 → 5 GND

PSU DATA

(SA30A-3400) ASTEC 230v



70w

+5v 8A
+12v 3.5A
-12v 1A
-5v 0.7A

220 - 240v @ 0.9A

/usr/egood/usr/lib/ell/wysonts/fonts
/usr/lib/ell/wyse/rgb.txt

UNIX

wdog = / (Root)
wdoh = /usr
Netscape Default Background
= COCOCOA.

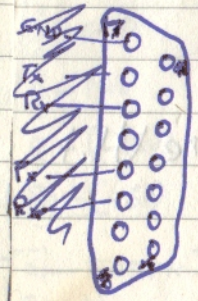
xterm Weware V1.1
Nov 21, 1994

Rev A

~~Bash~~/~~WARE~~

/usr/eware/arxC
/urs/eware/arxC-all

TTY Connector



8 - GND
7 - TX (Term 1)
6 - RX (Term 1)

4 - Tx (Term 2)
3 - Rx (Term 2)

Colour Code

8 (Black / Solid Blue)
7 (LIGHT GREEN / YELLOW)
6 (SOLID GREEN)
4 5 (LIGHT ORANGE / ORANGE)
3 (SOLID ORANGE / RED)

TTY MULTI WAY CABLE

10 - BROWN - LINK 2 (TERM TX)
9 - BLUE - LINK 2 (TERM RX)
SLIP/PPP/CSLIP

12 - LINK 3
13 - LINK 3

(TTY 2) / SLIP 2

SPECIAL SLIP ONLY CABLE

10 - ORN/WHT -
9 - ORANGE - LINK 2

13 - GRN / WHT -
12 - GREEN - LINK 3

ETHER ADDRESSES: CS = CLIENTS SERVER
AS = APPLICATIONS SERVER

CS(1) = ~~A0 FA 00 00 00 01~~

AS(2)/CLIENT = A0 FA 00 00 00 02

AS(1) = A0 FA 00 00 00 03

Wyse & Term (FLASH) 00:80:64:04:39:63

CS(1): 0 0 EB CC 67 A9

(SHAFER)
SUPER AS(1) = 00 00 EB CC 71 F9

(Rec)
Router S. AS(2) = 00:40:33:39:F3:04

Tracepace SN:

S# [redacted]
MODEL # URB2 TN
P# 90141-01

1.0
1.1

850 = 1651
16, 63

520 = 1049
MB
16, 63

12mg

800

Eet [redacted]

0172

COM 5.
4

BRNO	300	IRQ 10
ROUEN	300	IRQ 5
FANG 1	280	11
FANG 2	300	10

1.3 GB (MASTER)
2595, 16, 63